DOI: 10.56652/ejmss2024.1-2.3

Artykuł oryginalny / Original article

Piotr Godziewski
Faculty of Management, Wrocław University of Science and Technology, Wrocław, Poland, e-mail: piotr.godziewski@pwr.edu.pl

Critical infrastructure modernization: directions for services & sales transformation

Modernizacja infrastruktury krytycznej: kierunki transformacji w organizacjach serwisowych i sprzedażowych

ABSTRACT

Context: Large-scale software companies adopt continuous delivery (CD) practices widely. Short delivery cycles with frequent but small product changes reduce time-to-market and improve quality. Suppliers of critical network infrastructure products cannot exploit CD benefits.
Objective: This study shifted the research focus from the product development organization to the customer interface. It evaluated the understanding, knowledge, and opinions of the customers' delivery stakeholders about the implications of the CD transformation on the supplier and cus-tomer's way of working.
Method: A semi-structured in-depth interview protocol was developed around three research questions. Fifteen employees of the Finnish telecommunication vendor were interviewed.
Results: Considering extreme automation throughout the value stream, the commercial model and service offering must be adjusted for fast-paced delivery projects. Increasing the intensity of product updates required hardening of the deployment governance. Differences in CD readiness between products in the same solution portfolio hindered operational benefits.
Conclusions: Hot spots of CD transformation were identified to unblock CD operationalization at the customer interface. The CD North Star, which spells out benefits and new ways of working, would support the most sensitive change management activities such as revising product-attached services and creating new commercial models.

Keywords: continuous delivery, agile, DevOps, telecommunications, critical infrastructure.

STRESZCZENIE

Kontekst: Duże firmy programistyczne powszechnie stosują ciągłe dostarczanie (CD – continuous delivery). Krótkie cykle z częstymi, niewielkimi zmianami skracają czas wprowadzenia na rynek i poprawiają jakość. Dostawcy krytycznej infrastruktury sieciowej nie mogą jednak w pełni korzystać z zalet CD.
Cel: Badanie przesunęło uwagę z organizacji rozwijającej produkt na interfejs z klientem. Oceniono wiedzę, rozumienie i opinie interesariuszy klienta odpowiedzialnych za dostarczanie w kontekście wpływu transformacji CD na współpracę dostawcy i klienta.
Metoda: Opracowano półustrukturyzowany protokół wywiadów wokół trzech pytań badawczych. Przeprowadzono piętnaście wywiadów z pracownikami fińskiego dostawcy telekomunikacyjnego.
Wyniki: Przy skrajnej automatyzacji model biznesowy i oferta usług muszą być dostosowane do projektów o szybkim tempie. Częstsze aktualizacje produktów wymagały wzmocnienia nadzoru nad wdrożeniami. Zróżnicowany poziom gotowości do CD w portfelu rozwiązań ograniczał korzyści operacyjne.
Wnioski: Zidentyfikowano kluczowe obszary transformacji CD, których przezwyciężenie umożliwia wdrożenie CD na styku z klientem. Koncepcja CD North Star, określająca korzyści i nowe sposoby pracy, wspiera najbardziej wrażliwe działania zarządzania zmianą, jak rewizja usług produktowych i tworzenie nowych modeli biznesowych.

Słowa kluczowe: ciągłe dostarczanie, zwinne wytwarzanie oprogramowania (agile), DevOps, telekomunikacja, infrastruktura krytyczna.

otrzymano / received: 03.09.2024 | poprawiono / corrected: 15.10.2024 | zaakceptowano / accepted: 20.10.2024

DOI: 10.56652/ejmss2024.1-2.3

1. INTRODUCTION

1.1. CONTINUOUS DELIVERY

Continuous delivery (CD) is a recognized practice for managing the delivery of software products to customers and users. CD is characterized by frequent provisioning of small incremental product changes in short delivery cycles (Humble, 2018). The intention is to distribute risks over time, so that the potential failure of a single, but small, product change can be immediately addressed at a low cost (Chen, 2015). That advantage was particularly evident in relation to big-bang, one-time, waterfall delivery projects, where failure to roll out massive product changes often derailed the entire activity. A step forward from waterfall models was triggered by the rising volatility of customer requirements and the higher velocity at which those requirements were given to product development teams. Researchers promoted overlap between the design and implementation phases (Clark et al., 1995). The approach was extended by the concept of continuously monitoring signals, data, and information from customers and feeding them back to the product organization for more informed decisions (Noori et al., 1999). In the Agile Manifesto, the software engineering community prioritized customer collaboration over contract negotiation and responding to change over following plans.

The new paradigm of managing customer deliveries was reflected in two of the twelve guiding principles, i.e.,

- *"Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.",*
- *"Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale".*

Product organizations that successfully implemented CD at scale increased their competitiveness thanks to fast feedback loops supporting product experimentation (Deng et al., 2019). Many companies, such as Meta (Rossi et al., 2016) or Netflix (Lindon et al., 2022), recognizing the benefits of CD, made it part of release governance, pushing for tens or hundreds of atomic product changes weekly or even daily. On the other hand, CD transformation in highly regulated B2B (business-to-business) environments, such as banking, telecommunications, and automotive, proceeded more slowly (Rissanen & Münch, 2015), with specific challenges (van der Valk et al., 2018). This article examined the industrialization of CD for delivering critical network infrastructure products.

1.2. Critical network infrastructure

Policymakers define critical infrastructure as guidelines and requirements for market regulators, companies, service providers, and suppliers operating in the essential sectors of the economy. For example, Presidential Policy Directive/PPD-21: Critical Infrastructure Security and Resilience (2013) listed 16 sectors, including communications, emergency services, and information technology. Energy and communications systems were mentioned as enablers for all other sectors. (Critical Infrastructure Sectors, 2020) stated that the sixteen sectors were "(...) so vital to the United States that their incapacitation or destruction would have a debilitating effect on security, national economic security, national public health or safety, or any combination thereof". The European Union spoke similarly, recognizing that networks were the foundation of digital transformation. Therefore, Communication Service Providers (CSPs) were among the most critical business entities required to adhere to the highest regulatory, compliance, and reporting regimes. Critical infrastructure refers to assets, facilities, equipment, networks, and systems that enable essential services. The essential services were "crucial for maintaining vital societal functions, economic activities, public health and safety, or the environment." (NIS 2 Directive, 2022).

In each country, a telecom market regulator (e.g., the US Federal Communications Commission or the Polish Urzad Komunikacji Elektronicznej) keeps CSP accountable for complying with regulatory requirements and reporting regime. Strategies are created to protect critical infrastructure, both its physical and digital assets. In Poland, Narodowy Program Ochrony Infrastruktury Krytycznej became the focal point between government agencies involved in essential sectors. The program recognized the academic community as influential and emphasized the importance of research projects, including risk assessments, threat identification, interdependent modeling, and validation of best practices. It was one of the drivers behind publishing this work. Our focus was the supplier side of the B2B relationship, i.e., vendor delivering network function solutions to CSP. Building on the proven benefits of CD, our goal was to provide guidelines for transforming network infrastructure delivery projects to adopt CD. Such projects are executed jointly by the customer and supplier. Therefore, there was value in focusing on vendor organizations and their customer interface.

2. THEORETICAL BACKGROUND

Continuous delivery, an engineering practice rooted in software engineering, is not a universally defined term. Instead, we identified multiple continuous engineering practices. Continuous development and continuous testing were at the heart of execution for product teams operating in an Agile framework. Software engineers continuously checked in changes (the smaller, the better) to automatically validate them through the integration pipelines in build-test-deploy cycles (Kneuper, 2018). A quick feedback loop, in seconds or minutes, delivered immediate signals about whether the change was heading in the desired direction or failing. Continuous quality assurance requires a variety of test cycles with turnaround times that depend on the test scope (Contan et al., 2018). For example, system-wide verification could take anywhere from 24 hours (for daily feedback) to 48 hours (for product performance testing) to 72 hours (for product stability testing). When new software passed all testing levels, it would be considered ready for customers. Mature R&D organizations integrated continuous testing into release pipelines to reduce the likelihood of faults in the production environment (Penson et al., 2017). The ultimate step was deployment to the customer. Techniques, such as continuous deployment zones

(Dakkak et al., 2021), helped reduce deployment risks for mission-critical products.

We walked through continuous engineering practices in creating, releasing, and deploying new software. As mentioned earlier, continuous delivery was often described as a product creation paradigm that connects all activities, emphasizing that they are executed in short cycles, creating small outputs and thus minimizing risk at each phase. Industry practitioners have found benefits, including increased competitiveness through reduced time-to-market (Block et al., 2019), higher product quality, and adherence to coding guidelines (Vassallo et al., 2016), as well as improved productivity and collaboration (Itkonen et al., 2016). This work focused on product delivery phenomena at the vendor-customer interface. For this reason, we prioritized the commercial aspects of CD over the software engineering heritage. We understood commercial as the elements of sales, services, and portfolio management governance impacting customer delivery.

Bibliometric analysis revealed that the literature on CD primarily focused on engineering practices implemented by organizations that create products rather than those that deliver to customers or operate in a customer environment (Godziewski, 2023). The DevOps concept was closely related to CD and described a collaborative, multidisciplinary effort that enables the continuous delivery of high-quality software (Leite et al., 2019). The research originated primarily from the field of computer science. However, there were meaningful contributions from management and quality sciences. Thus, we studied several publications that propose frameworks for transforming people, process, and product.

The HURRIER model assumed the product to be an experimental system, continuously refined through testing customer value hypotheses (Issa Mattos et al., 2021). The case study was mission-critical telecommunications. It considered the limitations of the B2B setup and proposed that the type of experiment determines the choice of delivery technique. For example, a bug fix required a delivery framework quite different from the one suitable to validate novel business hypotheses as part of a new product introduction. For such experiments to occur continuously, the product architecture had to support them with the capability of producing small, atomic deliverables of sufficient quality for customer pilots (Bosch & Eklund, 2012). It was possible to create customer feedback loops as in the HYPEX model (Bosch, 2014) and ensure that product management decisions were data-driven and made based on customer feedback linked with purposeful experiments (Fabijan et al., 2015). The problem of collecting data from the installed base of mission-critical products was more complex. Tight dependencies between embedded software and underlying hardware platforms, B2B relationships, and regulatory constraints made implementing automated, continuous data collection mechanisms harder. Fabijan et al. (2016) explored solutions within the Feature Lifecycle Model, which connects the operationalization of customer feedback with product lifecycle stages. Bosch & Olsson (2016) further investigated product and organizational capabilities to establish a dynamic system loop that would enable the product to correct itself and adjust to changing customer value hypotheses. We identified stakeholder management and people communication as the distinct challenges in CD transformation. For example, the TAS model encouraged adjusting communication from product testing teams depending on the type of business stakeholders on the receiving side (Mårtensson et al., 2019). By doing that, testing organizations would increase stakeholders' trust and confidence in continuous testing.

We began by identifying the flavors of continuous delivery in terms of the continuous engineering practices adopted during the product creation process. Various definitions and subtle differences in interpretation (e.g., release vs. deployment, delivery vs. release) led to a situation where product organizations often developed company-specific definitions of CD. Therefore, the first research question aimed to establish the bridge between the academic version of CD, as defined earlier in this chapter, and the understanding of CD by industry practitioners operating at the customer interface.

RQ1: How do the supplier's customer interface stakeholders understand the concept of CD?

As we explored the industry-recognized and scientifically verified benefits of CD, we examined several CD models, including case studies involving mission-critical products. Models and frameworks addressed transformative impacts on people (e.g., collaboration), processes (e.g., frequency of deliveries), and products (e.g., architectural capabilities). Building on that research, we reached out to customer interface stakeholders to gather their views on the implications of CD transformation across supplier organizations. The key differentiation of our study was its focus on sales, services, and portfolio management, rather than R&D, which is already extensively covered in the CD literature.

RQ2: What are the implications for the supplier's way of working when industrializing CD at scale?

We found little to no reference to CD research developed directly with customers. In this study, we did not directly address the customer's voice. Still, sales and services representatives, especially those in the supplier's regional teams, were good proxies of the customer's voice. Access to organizations working directly with customers allowed us to ask questions about CSP expectations.

RQ3: What are the expectations of CSP customers when industrializing CD for critical network infrastructure solutions?

3. METHOD

We established that CD research was skewed toward product development organizations and processes. In this study, we investigated the voice of stakeholders at the customer interface of the supplier's organization in a B2B environment with CSP. The objective of RQ1 was highly descriptive and meant to establish a practical understanding and interpretation of the CD model in the case company. RQ2 and RQ3 supported an exploratory nature,

DOI: 10.56652/ejmss2024.1-2.3

diving into the prerequisites and potential consequences of CD transformation. Based on that mixture, we found the semi-structured in-depth interview method most appropriate (Wohlin et al., 2000). We categorized interview questions into four groups: knowledge and facts, opinions and values, experience, feelings (Hove & Anda, 2005). RQs drove the primary interview questions, while the secondary interview questions provided a bridge between the academic narrative and company-specific CD language known to the interviewees.

Each interview took between one and two hours. It began with a five- to ten-minute introduction outlining the academic objective and the business goal of commercializing CD at scale. Interviewees were informed that the study was part of the industrial PhD project, which had been approved and supported by the case company. The interviewer's role was that of a PhD student, rather than a job position held in corporate structures. All interviews were conducted remotely in English using Microsoft Teams, recorded, and transcribed. Auto-coding, manual coding, and thematic analysis were performed in NVivo Pro.

All fifteen interviewees were employed by the case company and listed in Table 1. We covered three market geographies, namely Europe, North America, and the Asia-Pacific, including Japan. There was a mix of professionals from global and regional divisions. Product and Commercial Managers responsible for all three markets belonged to the global group. Their decisions, such as product roadmap or pricing scheme, directly impacted the world-market customer operations. Sales Managers, Customer Program Managers, Account Managers, and Customer Success Managers comprised the regional teams responsible for customer relationships and commercial project delivery.

Table 1. Interviewees

Interview ID	Interviewee role	Scope of responsibility	Market geography
1	Product Manager	Global	Europe
2	Product Manager	Global	NAM
3	Product Manager	Global	APJ
4	Customer Program Manager	Customer account	NAM
5	Sales Manager	Customer account	NAM
6	Sales Manager	Customer account	Europe
7	Head of Presales	Customer account	Europe
8	Customer Program Manager	Customer account	NAM
9	Product Manager	Global	Europe
10	Sales Manager	Customer account	Europe
11	Account Manager	Customer account	Europe
12	Commercial Models Specialist	Global	Europe
13	Commercial Models Specialist	Global	Europe
14	Customer Success Manager	Customer account	APJ
15	Customer Success Manager	Customer account	APJ

Source: Own elaboration.

3.1. Case company

Nokia is a Finnish multinational telecommunications company. It operates through four business groups: Mobile Networks, Network Infrastructure, Cloud and Network Services, and Nokia Technologies. Nokia's customers span several sectors, reflecting the company's wide range of products and services. CSPs are the primary customers of mobile and fixed network infrastructure solutions. Enterprises across manufacturing, energy, transportation, and the public sector grow their share of wallet as they require private networks and digital transformation solutions.

The people, processes, and products we referred to in this study were associated with the CNS business group. It is the most software-centric division of Nokia. Its mission is to harness the power of cloud computing and advanced networking technologies through multiple diverse portfolios. We anchored this study in the Core Networks (CN) portfolio. It offers Virtualized Network Functions (VNF) and Cloud Native Functions (CNF), which provide networking capabilities such as Packet Core (data transfer) or IP Multimedia Subsystem (voice communications). CN products can be considered lane markers and traffic lights, prioritizing, routing, and optimizing information flow through critical network infrastructure. Whether it involves greenfield (new network function installation) or modernization (software upgrade, supplier swap, traffic migration), CN delivery projects carry high commercial and regulatory risks because each network function site serves a significant number of subscribers, i.e., hundreds of thousands to millions.

4. RESULTS

4.1. Understanding of the CD model

Our interviewees focused on expected CD benefits such as reduced time-to-market and accelerated customer acceptance milestones. Project cadence, understood as the frequency of product updates, was less concerning. Monthly updates were the target state. Quarterly updates provided a more realistic view of the intermediate phase. Multiple stakeholders advised us to adjust the CD project cadence based on the type of product deliverable. For example, a quick fix aimed at unblocking critical modernization projects should be provided on demand, ideally in an automated fashion, from the R&D CI pipeline through the digital supply chain to the customer environment. Other maintenance patches, such as security vulnerability fixes, could have their own regular, monthly cadence. One delivery manager stated that the frequency of major releases should determine the CD-based product refresh cadence. For example, continuous deliveries should come twice as often as major releases. If there were four major releases per year, the CD model should allow customers to absorb eight minor refreshes in the field.

We found the element of a feedback loop in the definitions shared by our respondents. They valued faster interaction and improved communication flow between their customers and product line organizations.

Automation was another common theme used to define CD. One person best captured the ambition toward extreme automation: "as touchless as possible for both regular deliveries as well as any emergency fixes, maintenance patches, or security updates". Interestingly, managers did not discuss the technological aspects of automation extensively; instead, they focused on the need to adapt systems and procedures along the end-to-end

delivery path. Those included digitization and automation of steps such as handling customer information questionnaires (CQI), creating product artifacts for deployment, authorizing the transfer of artifacts to customer and market regions, and finally rolling out the change to the lab or production.

4.2. Implications for the Vendor's way of working

One interviewee admitted that at some point in the past, "focus had not been on increasing the frequency of product deliveries, but rather on getting CD pipeline setup right" from the technology point of view. However, that was no longer the case.

Visibility, defined as the flow of information about the availability of product line deliverables, was one of the pressing challenges. Multiple participants urged the organization to modernize repositories used to exchange product artifacts with customers. Statements such as "delivery and deployment need to be looked at together" promoted the idea of a digital delivery catalog bridging R&D milestones (e.g., general availability, ready for piloting, ready for verification) with commercial gates (e.g., ready for lab deployment, restricted field deployment only, ready for mass rollout, critical quality alert). One expert outlined the vision of customers being empowered to select artifacts from such a catalog independently, without the need for the supplier's service personnel. All types of product deliveries, including major releases, maintenance packages, bug fixes, and security patches, would be channeled through a single digital catalog.

The interviewees from regional teams extensively discussed the solution-level approach to CD. In one dimension, it was about a portfolio, i.e., products that complement each other, and business applications grouped according to their underlying use case. Another dimension was the need for a platform to deploy business applications at scale. Platforms included cloud infrastructure, management systems, backup and restore functions, etc. Our participants warned of additional effort needed to address noticeable differences in CD capabilities between the elements of the same solution. It was a call for harmonizing CD capabilities across the portfolio and its associated platforms. Otherwise, CD benefits, especially operational savings from automation, were significantly limited or, in the worst case, diminished if one solution element fell largely behind others in its CD readiness.

Many comments referred to automation, specifically automation of customer acceptance procedures. One respondent described the vision of customer acceptance as "totally end-to-end, without changing the configuration of the system under test". There may have been a particular discrepancy between the expectations of sales professionals expressed during the interviews and the R&D strategies of the case company. Development organizations were excelling in product validation practices suited for the global market, but regional sales and services required explicit consideration of the customer-specific setups they had to deal with in projects.

Our interviewees were concerned about aligning commercial models with CD. One person explained that "CD technology framework and commercial model were not running at the same pace". For example, the same level of service engagement in high-frequency CD projects as in classical big-bang deliveries could easily derail the business case due to unacceptable service costs. Ideally, as the supplier organization progresses with CD automation, the delivery project's bottom line should have fewer cost items covering human-intensive, manual service efforts. The respondents followed with proposals to create high-value professional services in place of routine manual tasks. Some suggested pricing premiums for completely autonomous network operations.

4.3. Implications for the customer's WAY OF WORKING

All interviewees agreed that, in a highly regulated telecommunications industry, it was unrealistic for a product change to reach production without explicit approval from the customer. Some stakeholders hypothesized that joint customer-supplier services teams would support frequent, continuous deliveries. Such a team would have a limit of authority to approve the rollout of certain types of product changes, e.g., security updates. Team members could work in agile sprints dictated by the CD cadence of the coming product updates.

Services professionals told us that customers were looking for ways to harden the deployment governance. At that point, the concept of GitOps emerged in the discussions. The GitOps paradigm moves all product configuration artifacts to a version-controlled repository, just as we do with the software codebase. It solves the problem of protecting the system configuration, which relies on the CD pipeline (Ramadoni et al., 2021). One respondent described GitOps as follows: "You [network operations engineer] are not allowed to do any direct change using the [network] element manager or directly in the [business] application". Several interviewees acknowledged that GitOps operationalization should be closely tied to CD transformation, as this would provide customers with a comprehensive view of configuration change history and enable role-based access control, determining who can read, write, and execute a particular change.

When asked about forces supporting CD, our sales professionals categorized customer representatives into two groups. Operations personnel, i.e., those responsible for keeping things running, were reluctant to hear about CD if their prior experience with product upgrades was negative and reflected in comments such as "we never get product upgrades the first time right." On the other hand, people in planning and engineering, typically closer to the chief technology officer, were incentivized to introduce new features and innovations to gain a competitive advantage. That was the target group for initial engagements about CD transformation.

5. DISCUSSION

CD transformation received much attention from product development organizations, but its implications on the customer interface were not studied in detail. This work was built on feedback from fifteen managers and professionals involved in

DOI: 10.56652/ejmss2024.1-2.3

Table 2. Key findings

Research question	Key findings
RQ1 (understanding)	• The most frequently described benefit was the reduction of deployment time, • Most interviewees defined frequent deliveries as happening once or twice per quarter; monthly or bi-weekly deliveries were the outliers, • Feedback loop between the customer and product development organization was the common theme, independent of the interviewee's role and scope.
RQ2 (implications to vendor)	• Profitability would suffer if the commercial model was not adjusted for the cost of continuous project management (at the customer interface) and the cost of automation (in services or product line organizations), • Some product-attached services were becoming obsolete, • CD benefits were not always clear, • CD transformation required changes to the competence layout of services teams, • CD capabilities needed to be harmonized across products belonging to the same solution.
RQ3 (implications to customer)	• Customers always had the last word before pushing things to the field, • Introducing CD at scale required hardening the deployment governance, e.g., with GitOps, • Customers expected flexibility in selecting CD systems and tools, • Big-bang modernization projects were to be replaced with continuous change and project management, like Agile in product development.

Source: Own elaboration.

critical infrastructure projects for CSPs. It added value to the body of knowledge as we reached out directly to sales, services, and portfolio management teams.

Stakeholders were aligned on the definition of CD, but none of them referred to the corresponding marketing communications available at that time. Our professionals presented a pragmatic approach and did not break down the CD concept into its various elements, such as continuous integration, which is well-known to R&D organizations. End-to-end thinking and a solution mindset dominated all discussions. No salesperson called it a success unless R&D excelled in continuous engineering practices and released frequently, and there was a measurable improvement in the pace of customer deliveries. From a product capability perspective, it was evident that CD transformation required a balanced investment across the portfolio of products. Otherwise, realizing its full benefit in the field projects was impossible. In the worst-case scenario, market services teams had to fill in the gaps in the delivery process manually. The challenge was especially pronounced with business applications that relied on third-party cloud providers. Our stakeholders needed the application's delivery cadence and practices to align with those followed by cloud suppliers.

We established the need to tailor the case company's commercial frameworks to make them suitable for frequent, continuous deliveries, rather than big-bang delivery projects. Moreover, with the increasing amount of automation, less cost would be spent on suppliers' services while customers could increase their operational autonomy. How could that be monetized, i.e., flexibility through autonomous network operations, created an interesting future research direction.

5.1. Transformation hot spots

Driven by the goal of creating tangible recommendations for change leaders, we reviewed the findings with two additional stakeholders: a portfolio manager and an account manager. We did not alter the interview outcomes, but we gained insights on how to present the results in a corporate environment effectively. Instead of developing transformation cookbooks per organization, we focused on organizational interfaces. Our reviewers recognized that many critical changes required the buy-in and involvement of multiple teams. The interview data were explicitly collected from sales, services, and portfolio management, while some feedback also referred to product development. Those references defined four organizational types for drawing the transformation hotspots between them, as presented in Figure 1.

Central to all recommendations was the Continuous Delivery North Star. It was the strategy spelling out the how and the why of CD transformation, emphasizing the impact across all four organizations. We found it essential to identify strengths and opportunities, as well as weaknesses and threats, concerning organizations and their functions. For example, investment in product automation (by R&D) would need to be coupled with the corresponding competence development in services, so that automation can be fully exploited in projects and not countered by service engineers worried about their future. In the same spirit, sales would need to adjust their offers to account for automation and fewer manual tasks, while also accommodating more customer deliveries within shorter project cycles.

Portfolio managers and product development leaders were encouraged to review the horizontal and vertical capabilities of CD products. Business applications, cloud infrastructure, life cycle management tools, and auxiliary systems must be harmonized for CD maturity. Differences within a solution could diminish CD commercial benefits, as customers seek solutions rather than individual products.

Similarly, portfolio managers and services leaders must revisit the product-attached services portfolio. What was sellable in the era of big-bang delivery projects was not necessarily a viable sales item in the era of frequent, small deliveries augmented with extreme automation. On the other hand, there was an opportunity to shift toward sophisticated, high-value, professional services.

We recommended that the interlock between services and

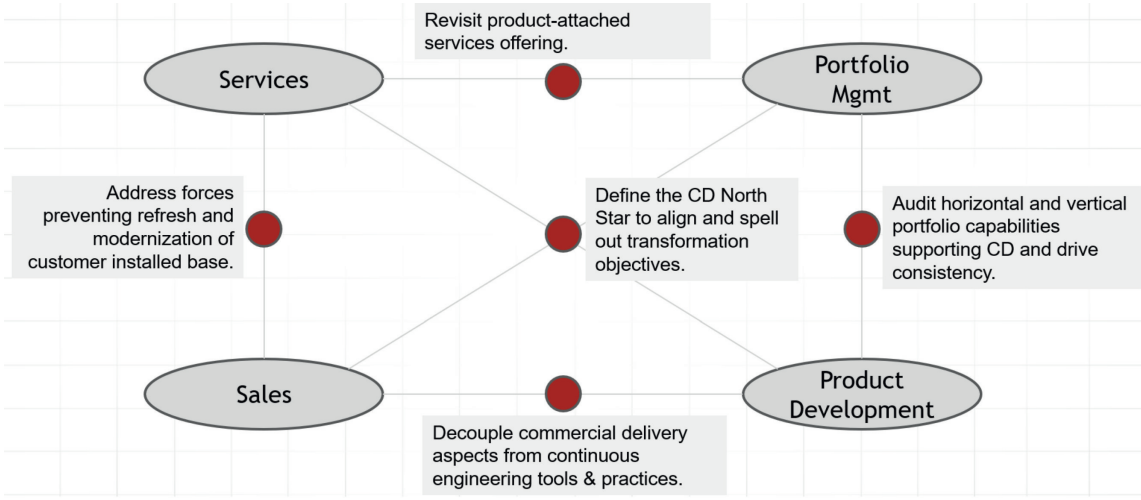

Figure 1. CD transformation hot spots at the organizational interfaces
Source: Own elaboration.

sales be enriched with incentives for the timely refresh and modernization of the customer's installed base. That would also improve the visibility of the installed base through continuous data collection and business intelligence analytics.

Finally, we advised sales and product stakeholders to decouple the commercial and engineering aspects of CD. R&D teams better handled technology requirements related to CD technologies, while sales managers worked on reshaping commercial frameworks (e.g., pricing), enabling a continuous project management office (e.g., joint customer-supplier DevOps teams), and advocating for CD transformation with the most influential customer representatives.

REFERENCES

Block, L., Riedel, O., & Herrmann, F. (2019). A lifecycle model to support continuous component evolution in embedded automotive systems. In M. Bargende, H.-C. Reuss, A. Wagner, & J. Wiedemann (Eds.), 19. *Internationales Stuttgarter Symposium: Proceedings* (pp. 1175–1189). Springer Vieweg. https://doi.org/10.1007/978-3-658-25939-6_94

Bosch, J. (2014). *Continuous software engineering*. Springer.

Bosch, J., & Eklund, U. (2012). Eternal embedded software: Towards innovation experiment sys-tems. In *Lecture Notes in Computer Science* (Vol. 7609, Part 1, pp. 19–31). Springer.

Bosch, J., & Olsson, H. H. (2016). Data-driven continuous evolution of smart systems. In *Proceed-ings of the 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems* (SEAMS 2016) (pp. 28–34). https://doi.org/10.1145/2897053.2897066

Chen, L. (2015). Continuous delivery: Huge benefits, but challenges too. *IEEE Software, 32*(1), 50–54. https://doi.org/10.1109/MS.2015.13

Clark, K., Henderson, R., Pisano, G., Pope, S., Smith, W., Tyre, M., von Hippel, E., West, J., & Iansiti, M. (1995). Shooting the rapids: Managing product development in turbulent environ-ments. *California Management Review, 38*(1).

Contan, A., Dehelean, C., & Miclea, L. C. (2018). Test automation pyramid from theory to prac-tice. In *2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)* (pp. 1–5). IEEE. https://doi.org/10.1109/AQTR.2018.8402699

Critical Infrastructure Sectors. (2020). U.S. Cybersecurity & Infrastructure Security Agency. https://www.cisa.gov/critical-infrastructure-sectors

Dakkak, A., Mattos, D. I., & Bosch, J. (2021). Perceived benefits of continuous deployment in software-intensive embedded systems. In *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)* (pp. 934–941). IEEE. https://doi.org/10.1109/COMPSAC51774.2021.00126

Deng, A., Gupta, S., Janowski, P., Kohavi, R., & Omhover, J. (2019). A/B testing at scale: Accelerating software innovation. In *Companion of the World Wide Web Conference (WWW 2019)* (pp. 1299–1300). ACM. https://doi.org/10.1145/3308560.3320093

Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union (NIS 2 Directive). (2022). *Official Journal of the European Union*. https://eur-lex.europa.eu/eli/dir/2022/2555/oj

Fabijan, A., Holmström, H., & Bosch, J. (2015). Customer feedback and data collection techniques in software R&D: A literature review. In *Lecture Notes in Business Information Processing* (Vol. 210, pp. 139–153). Springer. https://doi.org/10.1007/978-3-319-19593-3_12

Fabijan, A., Olsson, H. H., & Bosch, J. (2016). Time to say "Good bye": Feature lifecycle. In *Pro-ceedings of the 42nd Euromicro Conference on Software Engineering and Advanced Applica-tions (SEAA 2016)* (pp. 9–16). IEEE. https://doi.org/10.1109/SEAA.2016.59

Godziewski, P. (2023). Literature review of continuous delivery: Research directions for critical infrastructure software projects. Scientific Papers of Silesian University of Technology. *Organization and Management Series, 2023*(176). https://doi.org/10.29119/1641-3466.2023.176.7

Hove, S. E., & Anda, B. (2005). Experiences from conducting semi-structured interviews in em-pirical software engineering research. In *Proceedings of the International Software Metrics Symposium* (pp. 10–23). IEEE. https://doi.org/10.1109/METRICS.2005.24

Humble, J. (2018). Continuous delivery sounds great, but will it work here? *Communications of the ACM, 61*(4), 34–39. https://doi.org/10.1145/3173553

Issa Mattos, D., Dakkak, A., Bosch, J., & Olsson, H. H. (2021). The HURRIER process for experimentation in business-to-business mission-critical systems. *Journal of Software: Evolution and Process, 33*(9), e2390. https://doi.org/10.1002/smr.2390

Itkonen, J., Udd, R., Lassenius, C., & Lehtonen, T. (2016). Perceived benefits of adopting continuous delivery practices. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM 2016)*.

Kneuper, R. (2018). Software processes and life cycle models. *Cambridge International Journal of Computing and Digital Systems*.

Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2019). A survey of DevOps concepts and challenges. *ACM Computing Surveys, 52*(6), Article

DOI: 10.56652/ejmss2024.1-2.3

127. https://doi.org/10.1145/3359981

Lindon, M., Sanden, C., & Shirikian, V. (2022). Rapid regression detection in software deployments through sequential testing. In P*roceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '22)* (pp. 3336–3346). https://doi.org/10.1145/3534678.3539099

Mårtensson, T., Ståhl, D., & Bosch, J. (2019). Test activities in the continuous integration and delivery pipeline. *Journal of Software: Evolution and Process, 31*(4), e2153. https://doi.org/10.1002/smr.2153

Noori, H., Munro, H., Deszca, G., & McWilliams, B. (1999). Developing the "right" breakthrough product/service: An umbrella methodology – Part A. *International Journal of Technology Management, 17*(5).

Penson, W., Huang, E., Klamut, D., Wardle, E., Douglas, G., Fazackerley, S., & Lawrence, R. (2017). Continuous integration platform for Arduino embedded software. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)* (pp. 1–4). IEEE.

Presidential Policy Directive/PPD-21: Critical Infrastructure Security and Resilience. (2013). The White House Archives. https://obamawhitehouse.archives.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrastructure-security-and-resil

Ramadoni, Utami, E., & Fatta, H. Al. (2021). Analysis on the use of declarative and pull-based deployment models on GitOps using Argo CD. In *2021 4th International Conference on Infor-mation and Communications Technology (ICOIACT)* (pp. 186–191). IEEE.

Rissanen, O., & Münch, J. (2015). Transitioning towards continuous delivery in the B2B domain: A case study. In C. Lassenius, T. Dingsøyr, & M. Paasivaara (Eds.), *Agile Processes in Soft-ware Engineering and Extreme Programming (XP 2015)* (pp. 154–165). Springer. https://doi.org/10.1007/978-3-319-18612-2_13

Rossi, C., Shibley, E., Su, S., Beck, K., Savor, T., & Stumm, M. (2016). Continuous deployment of mobile software at Facebook (showcase). In *Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE 2016)* (pp. 12–23). https://doi.org/10.1145/2950290.2994157

van der Valk, R., Pelliccione, P., Lago, P., Heldal, R., Knauss, E., & Juul, J. (2018). Transparency and contracts: Continuous integration and delivery in the automotive ecosystem. In *Proceed-ings of the 40th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP 2018)* (pp. 23–32). ACM. https://doi.org/10.1145/3183519.3183543

Vassallo, C., Zampetti, F., Romano, D., Beller, M. M., Panichella, A., Di Penta, M., & Zaidman, A. (2016). Continuous delivery practices in a large financial organization. In P*roceedings of the 2016 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 519–528). IEEE. https://doi.org/10.1109/ICSME.2016.72

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2000). *Experimen-tation in software engineering*. Springer.